



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/690,214	10/21/2003	Michael J. Sundermeyer	47583/P042US/10311288	2346
59061 7590 05/31/2007 FULBRIGHT & JAWORSKI, LLP (ADOBE) 2200 ROSS AVENUE SUITE 2800 DALLAS, TX 75201-2784			EXAMINER LUDWIG, MATTHEW J	
			ART UNIT 2178	PAPER NUMBER
			MAIL DATE 05/31/2007	DELIVERY MODE PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

MAILED

MAY 31 2007

Technology Center 2100

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/690,214
Filing Date: October 21, 2003
Appellant(s): SUNDERMEYER ET AL.

Thomas J. Meaney
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed January 26, 2007 appealing from the Office action mailed September 8, 2006.

Art Unit: 2178

(1) *Real Party in Interest*

A statement identifying the real party in interest is contained in the brief.

(2) *Related Appeals and Interferences*

The examiner is not aware of any related appeals, interferences, or judicial proceedings, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) *Status of Claims*

The statement of the status of claims contained in the brief is correct.

(4) *Status of Amendments After Final*

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) *Summary of Claimed Subject Matter*

The summary of claimed subject matter contained in the brief is correct.

(6) *Grounds of Rejection to be Reviewed on Appeal*

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

The copy of the appealed claims contained in the Appendix to the brief is correct.

(7) *Grouping of Claims*

The appellant's groupings of the rejected claims are listed in the appeal brief.

(8) *Evidence Relied Upon*

GlobalScape, "CuteFTP Pro Technical Overview," White Paper, May 22, 2001 (hereinafter CuteFTP).

Art Unit: 2178

Altova Inc. & Altova GmbH, "XML Spy 4.0 Manual," September 10, 2001 (hereinafter XML Spy).

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-19 are rejected under 35 U.S.C. 102(b) as being clearly anticipated by GlobalScape, "CuteFTP Pro Technical Overview," White Paper, May 22, 2001, published on the internet at least by August 14, 2002, last downloaded by the Examiner on March 17, 2006, from:

<http://web.archive.org/web/20020814133509/www.globalscape.com/support/manuals.shtml>, pages i-ii and 1-17, [hereinafter "CuteFTP Pro"].

Regarding **dependent claim 1**, CuteFTP Pro teaches:

A method for maintaining a Web site comprising:

browsing to a Web page to be edited;

automatically downloading a source file for said Web page from a file transfer server related to said Web page including related files associated with display of said Web page;

(See, CuteFTP Pro, page 7, teaching: "CuteFTP Pro includes advanced scripting capabilities that allow clients to automate routine tasks, such as downloading log files from a Web server or posting weekly sales reports to an FTP server.")

editing said source file; and

Art Unit: 2178

(See, CuteFTP Pro, page 11, teaching an integrated code editor to change HTML/Text documents.)

automatically publishing said edited source file to said file transfer server associated with said Web site including said related files associated with said display of said Web page.

(See, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes.)

Regarding **dependent claim 2**, CuteFTP Pro teaches:

*The method of claim 1 further comprising:
scanning said Web page for page-dependent related files prior to said automatically downloading.*

(See, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated.)

Regarding **dependent claim 3**, CuteFTP Pro teaches:

*The method of claim 1 further comprising:
scanning said edited source file for modified page-dependent related files prior to said automatically publishing.*

(See, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated.)

Art Unit: 2178

Regarding **dependent claim 4**, CuteFTP Pro teaches:

The method of claim 3 wherein said modified page-related files comprises one or more of:

an edited page-related file;

a deleted page-related file; and an added page-related file.

(See, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated.)

Regarding **dependent claim 5**, CuteFTP Pro teaches:

The method of claim 4 further comprising:

translating local links to said added page-related files to reflect a location of said added page-related files on said file transfer server.

(See, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated. Note that CuteFTP Pro teaches folder synchronization, which reflects a location of added page-related files on the file transfer server.)

Regarding **dependent claim 6**, CuteFTP Pro teaches:

The method of claim 1 further comprising: mapping addresses of said edited source file and said related files associated with said display of said Web page to an address location commensurate with said file transfer server.

Art Unit: 2178

(See, CuteFTP Pro, page 8, teaching mirroring which makes the content of the remote drive exactly like the contents of the local drive, vice versa, or both.)

Regarding **dependent claim 7**, CuteFTP Pro teaches:

The method of claim 1 wherein said Web page is generated, in part, dynamically using embedded server code and a plurality of data stored in a database.

(See, CuteFTP Pro, page 11, teaching the use of a default template as embedded server code and customizable code options as data stored in the database.)

Regarding **dependent claim 8**, CuteFTP Pro teaches:

*The method of claim 7 further comprising:
identifying one or more elements generated from said database in said downloaded source file.*

(See, CuteFTP Pro, page 11, teaching the use of a default template as embedded server code and customizable code options as data stored in the database. Note also that CuteFTP Pro teaches to edit remote documents, which are identified elements from the database in the source file.)

Regarding **dependent claim 9**, CuteFTP Pro teaches:

The method of claim 8 further comprising:

checking said edited source file for modifications to said one or more elements.

((See, CuteFTP Pro, page 10, teaching that overwrite of files based on day and tie of last edit.)

Regarding **dependent claim 10**, CuteFTP Pro teaches:

The method of claim 9, wherein said automatically publishing comprises: stripping said modifications to said one or more elements from said edited source responsive to said checking; and updating said database with said modifications.

(See, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated. Note that CuteFTP Pro teaches folder synchronization, which reflects a location of added page-related files on the file transfer server.)

Regarding **dependent claim 11**, CuteFTP Pro teaches:

A computer program product having a computer readable medium with computer program logic recorded thereon for managing a Web site, said computer program product comprising: code for browsing to a Web page to be edited;

Art Unit: 2178

code for automatically retrieving from a file transfer server, one or more Web source files for at least one of: said Web page; and one or more Web page-dependent files;

code for editing said one or more Web source files; and

code for automatically uploading said edited one or more Web source files to said file transfer server.

(Claim 11 incorporates substantially similar subject matter as claimed in claim 1, and is rejected along the same rationale.)

Regarding **dependent claim 12**, CuteFTP Pro teaches:

The computer program product of claim 11 further comprising:

code for inspecting said Web page for said one or more Web page-dependent files prior to said code for automatically retrieving.

(Claim 12 incorporates substantially similar subject matter as claimed in claim 3, and is rejected along the same rationale.)

Regarding **dependent claim 13**, CuteFTP Pro teaches:

The computer program product of claim 11 further comprising:

code for examining said edited one or more Web source files for modified Web page-dependent files prior to said code for automatically uploading.

(Claim 13 incorporates substantially similar subject matter as claimed in claim 2, and is rejected along the same rationale.)

Art Unit: 2178

Regarding **dependent claim 14**, CuteFTP Pro teaches:

*The computer program product of claim 13 further comprising:
code for updating local links to one or more added Web page-dependent
files included in said edited one or more Web source files to remote links
reflecting file transfer server addresses of said one or more added Web page-
dependent files.*

(Claim 14 incorporates substantially similar subject matter as claimed in claim 6, and is rejected along the same rationale.)

Regarding **dependent claim 15**, CuteFTP Pro teaches:

*The computer program product of claim 11 further comprising:
code for translating addresses of said edited one or more Web source files
to an equivalent address for said file transfer server.*

(Claim 15 incorporates substantially similar subject matter as claimed in claim 5, and is rejected along the same rationale.)

Regarding **dependent claim 16**, CuteFTP Pro teaches:

*The computer program product of claim 11 wherein said Web page is
dynamically created, in part, using embedded server code in communication with
a database of information.*

(Claim 16 incorporates substantially similar subject matter as claimed in claim 3, and is rejected along the same rationale.)

Art Unit: 2178

Regarding **dependent claim 17**, CuteFTP Pro teaches:

The computer program product of claim 16 further comprising:

code for identifying one or more database elements in said Web page.

(Claim 17 incorporates substantially similar subject matter as claimed in claim 3, and is rejected along the same rationale.)

Regarding **dependent claim 18**, CuteFTP Pro teaches:

The computer program product of claim 17 further comprising:

code for detecting said one or more database elements modified during execution of said code for editing.

(Claim 18 incorporates substantially similar subject matter as claimed in claim 9, and is rejected along the same rationale.)

Regarding **dependent claim 19**, CuteFTP Pro teaches:

The computer program product of claim 18, wherein said code for automatically uploading comprises:

code for extracting said modified one or more database elements from said edited one or more Web source files; and

code for updating said database with said modified one or more database elements.

(Claim 19 incorporates substantially similar subject matter as claimed in claim 4, and is rejected along the same rationale.)

1. It is noted that any citations to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art.

See, MPEP 2123.

Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over Altova Inc. & Altova GmbH, "XML Spy 4.0 Manual," copyright 1998-2001, September 10, 2001, Chapter 3, consisting of pages 91-286, [hereinafter "XML Spy"]

Regarding independent claim 20, XML Spy teaches:

A method for managing a Web site lifecycle from a graphical user interface (GUI) comprising:

displaying a Web browser in a first window of said GUI, wherein a user browses on said Web browser to locate a Web page to be edited;

selecting an edit indicator displayed on said GUI indicating a preference to edit said Web Page;

responsive to said selecting, transitioning said first window to display a page editor;

choosing a publish indicator displayed on said GUI indicating to publish said edited Web Page; and

responsive to said selecting, transitioning said first window back to display said Web browser.

(See, XML Spy, pages 92-96, teaching the page window editor. See also XML Spy, page 93, teaching that the windows are freely selectable and may be clicked back and forth to bring them to the front.

XML Spy does not expressly teach wherein a user browses on the Web to locate a Web page to be edited. However, see, XML Spy, pages 264-267, teaching use of SML Spy with Internet Explorer and automatic notification of changes to a file located on a central server. It would have been obvious to one of ordinary skill in the art at the time of the invention to have used the function of XML Spy to monitor documents on a server, and to edit documents within a browser for a browser, to have used a browser to locate a Web page to be edited, for the obvious and beneficial purpose that XML Spy is obviously designed to cooperate with and use browsers in accessing and editing documents located on the Web.)

Claims 21-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Altova Inc. & Altova GmbH, "XML Spy 4.0 Manual," copyright 1998-2001, September 10, 2001, Chapter 3, consisting of pages 91-286, [hereinafter "XML Spy"] in view of GlobalScope, "CuteFTP Pro Technical Overview," White Paper, May 22, 2001, published on the internet at least by August 14, 2002, last downloaded by the Examiner on March 17, 2006, from:

Art Unit: 2178

<http://web.archive.org/web/20020814133509/www.globalscape.com/support/manuals.shtml>, pages i-ii and 1-17, [hereinafter "CuteFTP Pro"].

Regarding **dependent claim 21**, XML Spy in view of CuteFTP Pro teaches:

A Web page editor comprising:

a graphical interface (GUI) for receiving interaction from a user;

a Web browser displayed to said user in a main window of said GUI;

an edit indicator, associated with said display of said Web browser, presented to said user on said GUI, wherein said edit indicator controls execution of retrieval logic;

an edit screen replacing said Web browser in said main window responsive to said user selecting said edit indicator, wherein said user makes edits to said Web page; and

a publish indicator, associated with a display of said edit screen, presented to said user on said GUI, wherein said publish indicator controls execution of said upload logic.

(See, XML Spy, page 146-157, teaching the graphical user interfaces for controlling execution of retrieval logic, i.e.: identifying which documents to retrieve and from where.

XML Spy teaches the graphical user interface (GUI) with a selectable web browser or editor window, but does not teach a publish indicator to control execution of upload logic.

CuteFTP Pro teaches upload logic available either automatically or manually.

It would have been obvious to one of ordinary skill in the art at the time of the invention to have combined the editing and GUI functions of XML Spy with the upload logic of CuteFTP Pro for purposes of uploading XML and HTML files from the editor of XML Spy through the file transfer of CuteFTP Pro. The motivation to combine the references is taught in CuteFTP Pro in that it is designed to upload web compatible software to the web and web compatible software is taught to be created using XML Spy.)

Regarding **dependent claim 22**, XML Spy in view of CuteFTP Pro teaches:

*The Web page editor of claim 21 wherein said retrieval logic comprises:
code for analyzing said Web page for page-dependent related files.*

(See the rejection of claim 21, above, made applicable herein by this reference. See also, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated.)

Regarding **dependent claim 23**, XML Spy in view of CuteFTP Pro teaches:

*The Web page editor of claim 22 wherein said retrieval logic comprises:
code for automatically downloading a source file for said Web page from a
file transfer server related to said Web page; and
code for automatically downloading page-dependent related files.*

(See the rejection of claim 22, above, made applicable herein by this reference. See also, CuteFTP Pro, page 7, teaching: "CuteFTP Pro includes advanced scripting

Art Unit: 2178

capabilities that allow clients to automate routine tasks, such as downloading log files from a Web server or posting weekly sales reports to an FTP server.”)

Regarding **dependent claim 24**, XML Spy in view of CuteFTP Pro teaches:

*The Web page editor of claim 21 wherein said upload logic comprises:
code for checking said edited source file for modified page-dependent
related files prior to said automatically publishing.*

(See the rejection of claim 21, above, made applicable herein by this reference. See also, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated.)

Regarding **dependent claim 25**, XML Spy in view of CuteFTP Pro teaches:

*The Web page editor of claim 24 wherein said upload logic further
comprising one or more of:
an edited page-related file;
a deleted page-related file; and
an added page-related file.*

(See the rejection of claim 21, above, made applicable herein by this reference. See also, (See, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated.)

Art Unit: 2178

Regarding **dependent claim 26**, XML Spy in view of CuteFTP Pro teaches:

The Web page editor of claim 25 wherein said upload logic further comprising one or more of:

code for translating local links to said added page-related files to reflect a location of said added page-related files on said file transfer server.

(See the rejection of claim 21, above, made applicable herein by this reference. See also, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated. Note that CuteFTP Pro teaches folder synchronization, which reflects a location of added page-related files on the file transfer server.)

Regarding **dependent claim 27**, XML Spy in view of CuteFTP Pro teaches:

The Web page editor of claim 23 wherein said upload logic further comprising one or more of:

code for mapping addresses of said edited source file and said related files associated with said display of said Web page to an address location commensurate with said file transfer Server.

(See the rejection of claim 21, above, made applicable herein by this reference. See also, CuteFTP Pro, page 8, teaching mirroring which makes the content of the remote drive exactly like the contents of the local drive, vice versa, or both.)

Regarding **dependent claim 28**, XML Spy in view of CuteFTP Pro teaches:

The Web page editor of claim 23 wherein said Web page is generated, in part, dynamically using embedded server code and a plurality of data stored in a database.

(See the rejection of claim 21, above, made applicable herein by this reference. See also, CuteFTP Pro, page 11, teaching the use of a default template as embedded server code and customizable code options as data stored in the database.)

Regarding **dependent claim 29**, XML Spy in view of CuteFTP Pro teaches:

*The Web page editor of claim 28 wherein said upload logic further comprising one or more of:
code for identifying one or more elements generated from said database in said retrieved source file.*

(See the rejection of claim 21, above, made applicable herein by this reference. See also, CuteFTP Pro, page 11, teaching the use of a default template as embedded server code and customizable code options as data stored in the database. Note also that CuteFTP Pro teaches to edit remote documents, which are identified elements from the database in the source file.)

Regarding **dependent claim 30**, XML Spy in view of CuteFTP Pro teaches:

The Web page editor of claim 29 wherein said upload logic further comprising one or more of:

code for checking said edited source file for modifications to said one or more elements.

(See the rejection of claim 21, above, made applicable herein by this reference. See also, CuteFTP Pro, page 10, teaching that overwrite of files based on day and tie of last edit.)

Regarding **dependent claim 31**, XML Spy in view of CuteFTP Pro teaches:

The Web page editor of claim 29 wherein said automatically uploading comprises:

code for stripping said modifications to said one or more elements from said edited source responsive to said checking; and
code for updating said database with said modifications.

(See the rejection of claim 21, above, made applicable herein by this reference. See also, CuteFTP Pro, page 9, teaching to automatically update a web page every few minutes. Note that CuteFTP Pro scans the web page for the files to be updated. Note that CuteFTP Pro teaches folder synchronization, which reflects a location of added page-related files on the file transfer server.)

(10) Response to Argument

In response to appellant's argument regarding Independent claim 1 and 11 that the Examiner failed to address the limitation, 'browsing to a Web page to be edited...' the Examiner disagrees. CuteFTP Pro browses HTTPS sites using the same strong Security that Internet

Art Unit: 2178

Explorer/Netscape uses for secure e-commerce transactions. The CuteFTP reference states that FTP is the most robust way to exchange files between computers on the Internet. Like the Hypertext Transfer Protocol (HTTP), which transfers displayable Web pages and related files, and the Simple Mail Transfer Protocol, which transfers e-mail, FTP is an application protocol that uses the Internet's TCP/IP protocols. Finally, the CuteFTP offers the most advanced automation tools available in a Window's-based FTP application. Key features include: COM based scripting, folder synchronization, file transfer scheduling, Smart Overwrite logic, and an integrated code editor. All of the above-mentioned tools along with the CuteFTP Pro's Multiple Document Interface provide the user with a browsing means for locating/editing source files for web pages. Furthermore, the illustration on page 1 along with the HTTP methods provided would allow a user to download a source file for said web page that has been browsed to and downloads related files associated with display of said web pages. CuteFTP includes advanced scripting capabilities that allow clients to automate routine tasks, such as downloading log files from a Web server or posting weekly sales reports to an FTP server.

Also, in reference to independent claim 1 and 11, appellant states the CuteFTP reference fails to teach, 'automatically publishing said edited source file to said file transfer server associated with said Web site including said related files associated with said display of said Web page'. The section of CuteFTP titled 'Scheduling File Transfers' discloses a way to schedule file transfers through several options. One of those options is to update a web site every few minutes. Automatically updating a web page is a sufficient way to publish source files to the file transfer server associated with the web site including said related files associated with said display of said web page. Finally, regarding independent claim 1 and 11, appellant states

CuteFTP's file is neither a source file nor a file associated with display of a Web page browsed. The Examiner believes the reference teaches a means of updating web sites through file transfers. It should be noted that the CuteFTP file transfer program is used in a network environment. The updating is related to edited source files and associated with said display of said Web page. As presently claim the limitation fails to explicitly state how the related files are associated with said display of said Web page.

In reference to dependent claim 2 and 12, appellant believes the scheduling properties taught by CuteFTP which utilizes the continuous mirroring of a particular directory provides a means for scanning web pages for page-dependent related files prior to said automatically downloading. If a file was edited through the CuteFTP user interface and scheduled to update every 3 days than a scanning of the web page for page-dependent related files prior to said automatically downloading.

In reference to dependent claim 3 and 13, appellant believes the reference fails to teach 'scanning said edited source file for modified page-dependent related files prior to said automatically publishing'. CuteFTP Pro teaches a check of local files, scan, for changes to publishing. See CuteFTP, page 16, 'folder monitor'.

In reference to dependent claim 5 and 14, appellant believes the reference fails to teach 'translating local links to said added page-related files to reflect a location of said added page-related files on said files server'. Automatically updating a web page is a sufficient way to publish source files to the file transfer server associated with the web site including said related files associated with said display of said web page. The reference teaches a means of updating web sites through file transfers. It should be noted that the CuteFTP file transfer program is used

in a network environment. The updating is related to edited source files and associated with said display of said Web page. As presently claim the limitation fails to explicitly state how the related files are associated with said display of said Web page.

In reference to dependent claim 6 and 15, appellant states the merely mirroring an entire folder is not the same as mapping addresses of an edited source file and related files associated with the display of a Web page to an address location commensurate with a file transfer server. The reference to CuteFTP provides a method of mirroring which makes the content of the remote drive exactly like the contents of the local drive, vice versa, or both.

In reference to dependent claim 10 and 19, appellant asserts that CuteFTP does not disclose stripping modifications to one or more elements from an edited source responsive to a checking step and updating a database with the modifications. The Examiner would like to note the broad nature of the presently claimed language, 'stripping said modifications'. If the claim language is to be read broadly then the CuteFTP reference scanning the web page for the files to be updated and the folder synchronization, which reflects a location of added page-related files on the file transfer server, provides a means of stripping a modification and updating a web page based upon the update methods taught by CuteFTP.

In reference to independent claim 20, appellant argues improper motivation to modify XML Spy to meet the limitations of claim 20. The Examiner disagrees. XML Spy teaches the graphical user interface (GUI) with a selectable web browser or editor window, but does not teach a publish indicator to control execution of upload logic. CuteFTP Pro teaches upload logic available either automatically or manually. It would have been obvious to one of ordinary skill in the art at the time of the invention to have combined the editing and GUI functions of XML Spy

with the upload logic of CuteFTP Pro for purposes of uploading XML and HTML files from the editor of XML Spy through the file transfer of CuteFTP Pro. The motivation to combine the references is taught in CuteFTP Pro in that it is designed to upload web compatible software to the web and web compatible software is taught to be created using XML Spy.

Applicant merely alleges the lack of motivation, without further argument. XML Spy teaches the limitations specified in claim 20. XML Spy teaches downloading documents using a browser to locate a Web page to be edited. XML Spy does not expressly teach a user browsing the Web to locate a Web page to be edited. It was within the knowledge of one of ordinary skill in the art at the time of the invention to use a browser to download a file. CuteFTP Pro browses HTTPS sites using the same strong Security that Internet Explorer/Netscape uses for secure e-commerce transactions. The CuteFTP reference states that FTP is the most robust way to exchange files between computers on the Internet. Like the Hypertext Transfer Protocol (HTTP), which transfers displayable Web pages and related files, and the Simple Mail Transfer Protocol, which transfers e-mail, FTP is an application protocol that uses the Internet's TCP/IP protocols. Finally, the CuteFTP offers the most advanced automation tools available in a Window's-based FTP application. Key features include: COM based scripting, folder synchronization, file transfer scheduling, Smart Overwrite logic, and an integrated code editor. All of the above-mentioned tools along with the CuteFTP Pro's Multiple Document Interface provide the user with a browsing means for locating/editing source files for web pages. Furthermore, the illustration on page 1 along with the HTTP methods provided would allow a user to download a source file for said web page that has been browsed to and downloads related files associated with display of said web pages. CuteFTP includes advanced scripting capabilities that allow clients to automate

routine tasks, such as downloading log files from a Web server or posting weekly sales reports to an FTP server.

In reference to claim 21, appellant believes the references fail to disclose or suggest a web browser. . It was within the knowledge of one of ordinary skill in the art at the time of the invention to use a browser to download a file. CuteFTP Pro browses HTTPS sites using the same strong Security that Internet Explorer/Netscape uses for secure e-commerce transactions. The CuteFTP reference states that FTP is the most robust way to exchange files between computers on the Internet. Like the Hypertext Transfer Protocol (HTTP), which transfers displayable Web pages and related files, and the Simple Mail Transfer Protocol, which transfers e-mail, FTP is an application protocol that uses the Internet's TCP/IP protocols. Finally, the CuteFTP offers the most advanced automation tools available in a Window's-based FTP application. Key features include: COM based scripting, folder synchronization, file transfer scheduling, Smart Overwrite logic, and an integrated code editor. All of the above-mentioned tools along with the CuteFTP Pro's Multiple Document Interface provide the user with a browsing means for locating/editing source files for web pages. Furthermore, the illustration on page 1 along with the HTTP methods provided would allow a user to download a source file for said web page that has been browsed to and downloads related files associated with display of said web pages. CuteFTP includes advanced scripting capabilities that allow clients to automate routine tasks, such as downloading log files from a Web server or posting weekly sales reports to an FTP server.

In reference to dependent claim 22, appellant asserts that merely performing directory mirroring of an entire folder is very different from scanning a Web page for page dependent files. As presently claimed, the language, 'code for analyzing said web page' is broad in nature

and taught by the CuteFTP reference. The reference provides a web page that is scanned through the use of code for the files to be updated.

In reference to dependent claim 23, appellant states that CuteFTP fails to teach or suggest that its scripting feature is able to scan edited source file for modified page-dependent files, and the Examiner has not shown otherwise. The CuteFTP Pro, page 10, teaches that it will backup or download, “your entire site.” Any page-dependent files would be within the definition of “entire site” and, therefore, would also be downloaded. Code would be present for the above-mentioned tasks to take place and therefore, the CuteFTP reference teaches a method for allowing for code to automatically downloading page-dependent related files.

In reference to dependent claim 24, appellant asserts that CuteFTP only discloses mirroring a directory and that it does not translate local links to added page-related files to reflect a location of those files on a file server. The Examiner is unclear as to how the argument relates to the claim, which states, “code for checking said edited source file for modified page-dependent related files prior to said automatically publishing”. The Examiner cannot locate the specific language presented in the appellant’s argument in dependent claim 24. See, Cute, FTP Pro, page 10, “Folder Monitor,” teaching that a folder structure is checked prior to uploading. Any page-dependent files would be within the definition of “entire site” and, therefore, would also be checked.

In reference to dependent claim 26, appellant asserts that CuteFTP only discloses mirroring a directory and that it does not translate local links to added page-related files to reflect a location of those files on a file server. See, Cute, FTP Pro, page 10, “Folder Monitor,” teaching that a folder structure is checked prior to uploading. The program knows the locations of both

Art Unit: 2178

the local and remote files, for the comparison, and it would be obvious to add the links to the page-related files on the transfer server along with publishing the rest of the files.

In reference to dependent claim 27, appellant asserts that merely mirroring an entire folder is not the same as mapping addresses of an edited source file and related files associated with the display of a Web page to an address location commensurate with a file transfer server. The Examiner believes the reference to CuteFTP and the mirroring techniques taught, do provide the code necessary for mapping addresses of said edited source file and said related files associated with said display of said web page because the mirroring technique makes the content of the remote drive exactly like the contents of the local drive, vice versa, or both. This method provides a way to map addresses of an edited source file through the utilization of the browser methods taught by CuteFTP.

In reference to dependent claim 31, appellant states the Examiner failed to properly address the claim and rejects it on the same basis as the rejection of claim 26. The language may be different, however, that only, should not prevent the Examiner from utilizing the same method from rejecting a claim. Broad language could, alone, be a sufficient reason to use one method to reject two claims, three claims, four claims, etc. Appellant asserts that CuteFTP does not disclose stripping modifications to one or more elements from an edited source responsive to a checking step and updating a database with the modification. The reference discloses automatically updating a web page every few minutes. Note that CuteFTP scans the web page for the files to be updated. CuteFTP teaches folder synchronization, which reflects a location of added page-related files on the file transfer server.

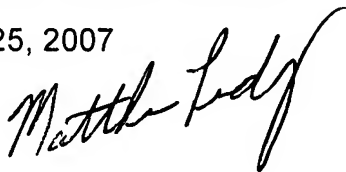
For the above reason, it is believed that the rejections should be sustained.

Respectfully submitted,

Matthew Ludwig

Assistant Patent Examiner

May 25, 2007



STEPHEN HONG
SUPERVISORY PATENT EXAMINER

Conferees

Stephen Hong

Supervisory Patent Examiner

Heather Herndon

Supervisory Patent Examiner



WEILUN LO
SUPERVISORY PATENT EXAMINER